

# PynPoint Code for Exoplanet Imaging

Adam Amara, Sascha P. Quanz, Joel Akeret

ETH Zurich, Department of Physics, Wolfgang Pauli Strasse 27, 8093 Zurich, Switzerland

---

## Abstract

We announce the public release of PynPoint, a Python package that we have developed for analysing exoplanet data taken with the angular differential imaging observing technique. In particular, PynPoint is designed to model the point spread function of the central star and to subtract its flux contribution to reveal nearby faint companion planets. The current version of the package does this correction by using a principal component analysis method to build a basis set for modelling the point spread function of the observations. We demonstrate the performance of the package by reanalysing publicly available data on the exoplanet  $\beta$  Pictoris b, which consists of close to 24,000 individual image frames. We show that PynPoint is able to analyse this typical data in roughly 1.5 minutes on a Mac Pro, when the number of images is reduced by co-adding in sets of 5. The main computational work parallelises well as a result of a reliance on SciPy and NumPy functions. For this calculation the peak memory load is 6Gb, which can be run comfortably on most workstations. A simpler calculation, by co-adding over 50, takes 3 seconds with a peak memory usage of 600 Mb. This can be performed easily on a laptop. In developing the package we have modularised the code so that we will be able to extend functionality in future releases, through the inclusion of more modules, without it affecting the users application programming interface. We distribute the PynPoint package through the central PyPi sever, and the documentation is available online (<http://pynpoint.ethz.ch>).

**Keywords:** methods: data analysis, techniques: image processing, planets and satellites:detection

---

## 1. Introduction

The field of exoplanet research has grown enormously in recent decades, with the current census of confirmed exoplanets and exoplanet candidates exceeding 5,000 objects (see, [www.exoplanet.eu](http://www.exoplanet.eu) [1], [www.exoplanets.org](http://www.exoplanets.org) [2]). The vast majority of the exoplanets known today have been detected through radial velocity (RV) and transit measurements (e.g., [3, 4, 5, 6, 7]). These studies, and in particular the *Kepler* space mission, allow us to derive robust statistics for the occurrence rate of planets on close-in orbits as a function of planet size and mass (e.g., [8, 9, 10, 5]).

In addition to RV and transit searches for exoplanets, a considerable and increasing effort has been devoted to direct imaging searches for exoplanets in recent years. These activities are currently culminating in the commissioning of dedicated high-contrast planet imaging instruments at 8-m class telescopes (SPHERE at the VLT [11], and GPI at Gemini [12]). The scientific motivation for direct imaging searches is at least twofold: (1) determining the occurrence rate of exoplanets at larger orbital separations not accessible by RV and transit searches; (2) obtaining spectra and spectral energy distributions (SEDs) of extrasolar planets to constrain their atmospheric parameters. In principle, exoplanet atmospheres can also be studied using transit and secondary eclipse observations (e.g., [13]), but only a small fraction of the exoplanet population is transiting in front of or behind their host star. Examples for directly imaged exoplanets that likely formed from a circumstellar disk around their host star include  $\beta$  Pictoris b [14], the 4-planet system around HR8799 [15, 16], HD95086 b [17] and GJ504 b [18]. In addition, in a few cases, there is observational evidence from direct imaging campaigns for very young planets that are embedded in or interact with the gas rich disk of their host stars (e.g. HD100546 [19]).

The biggest challenge for the direct detection of an exoplanet is the stark brightness contrast between the planet and its host star in combination with the small on-sky separation. Hence, direct planet searches are typically done with ground-based, adaptive optics assisted, very large telescopes at near-infrared (NIR) wavelengths ( $1 - 5 \mu\text{m}$ ), where one aims at the detection of the planet's thermal emission rather than reflected starlight. To enhance the

contrast capabilities of the instruments, coronagraphs with different designs have been introduced in the past years, in particular for observations between  $3 - 5 \mu\text{m}$  [21, 22, 23]. Furthermore, a big improvement in contrast performance was achieved with the introduction of the angular differential imaging (ADI) technique [24], which allows for a much more accurate subtraction of the central star’s point spread function (PSF).

Together with these technical improvements, it has also been realised that there are significant potential gains from developing better and more sophisticated data analysis methods. A particularly crucial step in the data processing chain is the modelling and subtraction of the star’s PSF, as typically the star outshines a planetary companion by orders of magnitude. Hence, only an accurate subtraction of the stellar flux contribution might reveal the existence of a nearby companion. A nice example here was the introduction of the LOCI algorithm [25] that was developed for the Gemini Deep Planet Search [26] and is still commonly used today. More recently, principal component analysis (PCA) based algorithms were introduced by our group and a group in the US [27, 28]. As the quality and volume of exoplanet direct imaging data will continue to increase in the coming years, in particular with the advent of the dedicated instruments mentioned above, the analysis methods used to process the data will also have to grow in sophistication.

In this paper, we give a brief overview of the PynPoint package. We begin with a brief description of the data and processing steps that currently need to be performed before, the data is passed to PynPoint (section 2). In section 3, we give an overview of the PynPoint package; and we describe the main user interfaces in section 4. We show performance results in section 5, and in sections 6 and 7 we present a discussion of quality assurance and a summary.

## 2. The Data and Initial Data Reduction Steps

Before describing in detail the functionalities of the PynPoint package, it is worth laying out the typical input data that PynPoint uses and the basic data reduction and analysis steps that are carried out before the PSF fitting and subtraction are performed.

As described above, the standard high-contrast observing technique to image exoplanets is ADI on adaptive optics-assisted NIR cameras. Typically during this process, the pupil of the telescope is fixed throughout the observing sequence, but the stellar field rotates on the detector due to the continuously changing parallactic angle caused by Earth’s rotation. Depending on the observing wavelength, the brightness of the star and whether or not a coronagraph is used, the typical exposure time of a single image frame varies between a fraction of a second to several seconds. A typical size of an individual raw image is  $1024 \times 1024$  pixels. However, in some cases smaller detector sub-arrays are read out and stored (e.g.,  $512 \times 512$  or  $256 \times 256$  pixels). A typical observing sequence lasts for up to a few hours, resulting in thousands if not tens of thousands of individual exposures that are stored individually. To be able to correct for bad pixels and sample the PSF more accurately, while properly subtracting background emission, the star is made to cycle through different pre-defined positions on the detector during the observing sequence (‘dithering’). The number of exposures taken at a given dither position before the star is moved to the next point depends on the observing wavelengths, the individual exposure time and the observing conditions. While observing at  $1 - 2.5 \mu\text{m}$ , one can normally spend up to several minutes at the same dither position. At longer wavelengths ( $3 - 5 \mu\text{m}$ ) and with more strongly varying background (see below), one switches to another dither position every 30 – 60 seconds.

The raw images undergo several steps before they are processed the PynPoint package. These process include:

**Parallactic angle determination.** Depending on the instrument, not all individual raw images will have a parallactic angle assigned to them. In that case, the parallactic angle for a given raw image has to be computed and stored in the header of the image file. PynPoint will search for a header keyword called `NEW_PARA` in each individual input image to access the parallactic angle.

**Dark current subtraction.** The dark current (and, where applicable, also the bias level) for a given exposure time and detector readout scheme is subtracted from all raw images. This step can be skipped if in the background subtraction step (see below) two individual images are subtracted from each other, as this will also subtract any dark current signal.

**Bad pixel cleaning.** Dead or bad detector pixels, as well as pixels possibly hit by a cosmic ray, have to be identified. These pixels are either masked out and disregarded from further analyses or they are replaced with the mean value of surrounding pixels.

**Flatfielding.** A flat-field correction is applied to the individual images to correct for large scale sensitivity gradients across the detector. In some cases, depending on the required calibration accuracy, this step can be skipped if the detector is reasonably ‘flat’.

**Background subtraction** Depending on the wavelength, the background emission has to be subtracted from all individual images. This step becomes more important with increasing wavelength and is absolutely crucial for data taken between  $3 - 5 \mu\text{m}$ . At these wavelengths, thermal emissions from the sky-background, the telescope and the instrument contribute to the total background flux. Without a nearby bright star, observations at  $3 - 5 \mu\text{m}$  are typically limited by the poisson noise of the background flux and temporal changes therein. To subtract the background emission from a given raw image, typically another image observed close in time but at a different dither position is subtracted. Alternatively, a sequence of several images from a different dither position is mean combined, to reduce the poisson noise in the background, and the resulting average image is removed. It has been suggested, that additional sensitivity can be gained if the LOCI algorithm is used to construct and subtract the background emission in the immediate vicinity of a star for a given input image [29].

**Image alignment and trimming.** As the star was recored at different dither positions on the detector, the individual images have to be aligned to a common centre. PynPoint requires that all input images are centred on the central star. In order to compute the spatial offset between the different images, either a 2-D Gaussian fit is used to determine the location of the photo-center of the stellar PSF, or the images are cross-correlated. Once the offset is known, the images are scaled up by a factor of at least ten, shifted and then re-scaled back to either their original size or twice that value. In PynPoint, we normally work with images that are up-scaled by a factor of two [27] as compared to the original pixel scale. This allows us to recover sub-pixel information on the PSF shape. Depending on the data set and the science goals, we typically have image sizes between 1 and 5 arcsec<sup>2</sup>.

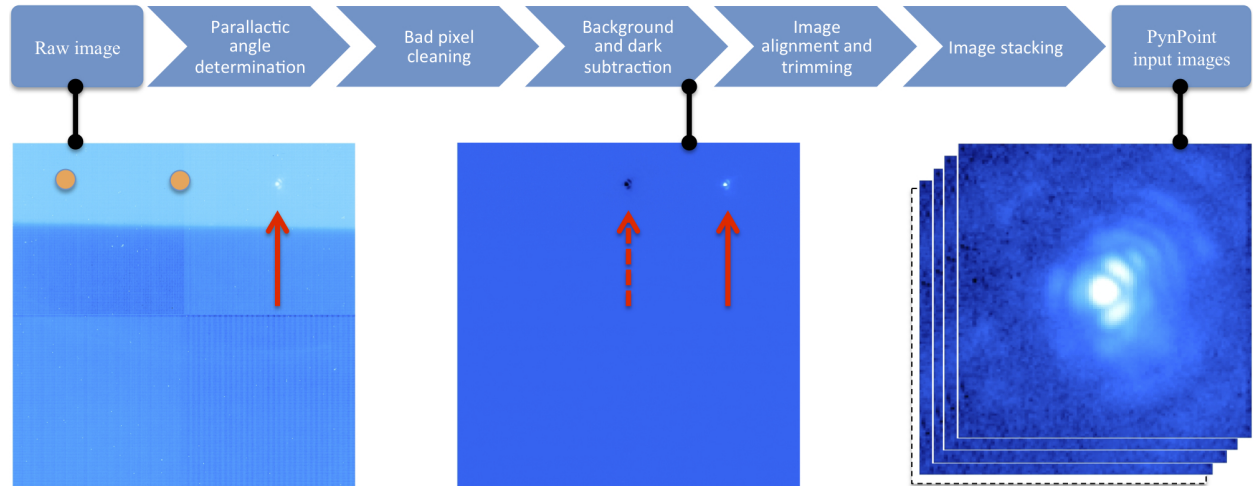


Figure 1: Initial data reduction steps for high-contrast imaging data taken with VLT/NACO in the  $L'$  filter. In this example, no flat-fielding is applied and the background emission and dark current contribution are eliminated via pairwise subtraction of individual images taken at different dither positions. Also shown is a single raw image (left), an intermediate image after background and dark subtraction step (middle), and a representation of a stack of final input images for PynPoint (right). The raw image is  $512 \times 512$  pixels in size ( $\sim 13'' \times 13''$ ). The high-background level, as well as the four detector quadrants are clearly visible. Also clearly seen is the bright stripe in the upper quarter of the image, which is related to the design of the APP coronagraph [21]. The current location of the star is indicated by the red arrow, and the other dither positions used in this observing sequence are indicated by the orange circles. In the intermediate image, the detector artefacts are gone and the background appears smooth. In addition to the star (red arrow), a negative imprint of the star is also visible (red, dashed arrow), resulting from the pairwise subtraction procedure. The final input images were created from a mean combination of 10 individual exposures after applying all initial reduction steps. The images are  $146 \times 146$  pixels in size ( $\sim 2'' \times 2''$ ) and were scaled up by a factor of two with respect to the original image size. The asymmetric PSF of the APP is clearly visible with the diffraction rings being strongly suppressed on the lefthand side of the central star. However, this image also demonstrates that in order to reveal faint companions next to the central star, the stellar PSF needs to be modelled and subtracted as accurately as possible. PynPoint usually processes thousands of such images.

For different instruments or observing strategies, the exact sequence of the steps listed above may vary slightly and not all steps maybe necessary. In the end, however, PynPoint (v0.2.0) expects a stack of individual .fits files that have all been pre-processed, with the same size in pixels and are aligned properly with the star sitting in the center of the image.

In Figure 1, we show a typical flow diagram of the initial reduction steps for the data [22, 30, 31, 19]. These data are taken with the NACO camera at the Very Large Telescope of the European Southern Observatory (ESO) in the L' filter (central wavelength  $\lambda_{\text{cen}} = 3.8\mu\text{m}$ ). In addition to the data reduction steps, Figure 1 also shows a single raw image, an intermediate data product and the final input images for PynPoint for illustrative purposes. In this particular case, the APP coronagraph [21, 22] was used, giving the PSF its asymmetric shape. It has to be emphasised that in each of the recorded images, the PSF of the central star has a slightly different intensity distribution on the detector due to continuously changing observing conditions and related variations in the AO corrections. This variability in the PSF in combination with variations in the uncorrected AO-halo and quasi-static speckles due to non-common path aberrations is the main challenge for PSF-fitting algorithms.

### 3. Package Description

PynPoint is a Python-based package that we have developed for processing high-contrast imaging data. In this first release, we have focused on the steps associated with modelling and subtracting the PSF of the bright central star. However, we have structured the package in a modular way so that extra features can be later added to the package easily. Once new features are developed and tested, we plan to make new releases publicly available. Therefore, an important constraint of the structure is that the released code can be scaled up in a way that maintains backward compatibility.

Version 0.2.0 of PynPoint contains the modules summarised in Table 1. Among these, the main three classes that will be accessed by the users are `basis`, `images` and `residuals`. These classes are used to generate an appropriate basis set for the analysis, treat the data (including modelling the PSF in each frame) and then to subtract the PSF model from each image and average the residuals through the stack of images.

In order to maintain a clean design, flexible modularity and a clear separation of concerns, we have created a series of modules to manage common features: functionality that is shared amongst multiple classes, such as saving, plotting and animating, has been grouped together in a class hierarchy. In order to facilitate the instantiation and restoration of these classes, a factory module has been developed. Additional generic functionality for manipulating the data has been defined in the `_Util` helper module.

An important part of the data analysis is to be able to mask regions of the image. The pixels at the centre of the PSF, i.e. the position of the star, are typically saturated. This means that their response to the input flux is far from linear. Hence, the central region is difficult to model accurately, and the information content in this area is minimal to none. Inclusion of these regions can, therefore, degrade the quality of the image analysis. For instance, if the basis set used to model the PSF is orthonormal, inclusion of the central region will affect the basis functions outside the central region. A simple and effective way of dealing with such problems is to mask unwanted regions. Masking operations in PynPoint are handled by the `_Mask` module.

As well as an interactive mode, discussed further below, we also offer the capability to run PynPoint in a sequence that can be predefined by the user. To manage these operations, we have developed a simple workflow engine (called `workflow`) that can read the user inputs and run the appropriate functions in the right order.

Detailed instructions of how to download and install the PynPoint package can be found on the package website<sup>12</sup>. The simplest method is to use `pip`<sup>3</sup>:

```
$ pip install PynPoint-exoplanet --user
```

This will install the PynPoint package and all of the needed dependencies, which are: NumPy, SciPy, PyFITS, matplotlib and h5py.

---

<sup>1</sup><http://pynpoint.ethz.ch>

<sup>2</sup><http://readthedocs.org/projects/pynpoint>

<sup>3</sup><http://pip-installer.org/>

Table 1: Main PynPoint modules.

Module	Description
Workflow	Engine for managing a set of PynPoint operations
Basis	Generates the basis set used to model the PSF
Images	Manages and prepares the images to be analysed
Residuals	Calculates and manipulates the residuals between image and model
PynPlot	Plotting and animation routines
.BasePynPoint	Set of base features used in basis, images and residuals
.Creators	Factory module
.Ctx	For shared resources, such as data, during session
.Mask	Provides functions for the masking operations
.Utils	Generic functionality for manipulating the data

#### 4. Application Programming Interface

The PynPoint package can be run in different modes that we outline in the sections below. The starting point assumes that there is a directory containing a set of fits files that have already been processed through the pre-PynPoint steps highlighted in Section 2. The .fits files should contain the images centered on the star and each image header should contain a parameter NEW\_PARA, which indicates the parallactic angle of this particular image. Example scripts for running PynPoint can be found in the examples folder of the package.

##### 4.1. Using Interactive mode

PynPoint can be run from the Python shell using keyword arguments to specify the calculations that should be performed by each module. See Table 2 for a list of the keyword options that can be passed to the classes images and basis. A typical session can be executed as follows:

```
>>> import PynPoint
>>> images = PynPoint.images.create_wdir(dir1,cent_size=0.05,stackave=5)
>>> basis = PynPoint.basis.create_wdir(dir2,cent_size=0.05,stackave=5)
>>> res = PynPoint.residuals.create_winstances(images, basis)
```

It is possible to create a basis instance with some of the keyword options different from those used to produce images. In general, this is likely not to be a sound idea and should be avoided. Therefore, users should take some caution to understand what they are trying to test if they use different options. In the example above, an instance of the image class is created by reading in data from the directory ('dir1'). This option reads in all the files in this directory that end in .fits and stores all the images in the array `im_arr`, which will have dimensions  $N_x \times N_y \times N_{files}$ , where  $N_x$  and  $N_y$  are the number of pixels in the x and y directions of each image and  $N_{files}$  is the number of files in the directory. We find that reading in a large number ( $\sim 10,000$ ) of small fits files can be very time consuming. Hence, instead of reading these files each time a particular directory is targeted, PynPoint converts the data into an HDF5 file<sup>4</sup>. This file is stored inside the original target directory and is given the same name as the directory with the hdf5 extension. In future calls, if the data in this directory is needed, PynPoint first checks to see if such a HDF5 file exists in the directory. If it does, then the data is loaded directly from the HDF5 file instead of the individual files.

The images stored (in the attribute `im_arr`) are: (i) normalised so that the sum of the pixel values is one; (ii) resized/re-centered (optional); and (iii) masked. The operations are managed by the function `prep_data` in the `.Util` module. All of the information needed to reverse these steps and to recover the original input data is contained in the `images` instance.

<sup>4</sup><http://hdfgroup.org/HDF5>

Option	Default	Description
recent	False	If True, the images will be re-centered.
resize	False	If True, the final images will be increased by a factor set by F_final.
cent_remove	True	If True, the central region will be masked (size set by cent_size).
para_sort	True	If True, the images will be sorted so that the parallax angles increase through the stack.
F_final	2	Factor increase in resolution of final images (resize must be True).
cent_size	0.05	Radius of the central mask as fraction of the full image size.
edge_size	1.0	Diameter of the outer mask in fraction of the image size. (1.0 corresponds to circle that touches the edge of the image).
stackave	None	If set to an integer N, then the stack will be reduced by averaging over adjacent N images.
ran_sub	None	If set to an integer M, then a random subset of this size will be analysed.

Table 2: Main keyword options used by `images` and `basis` along with their default values.

The `basis` class is designed to generate and manage the basis set that will be used to model the PSF. In the current version, this is done by constructing a set of principal components using singular value decomposition (SVD). For details on the performance of this basis set, see [27, 32]. However, we have structured the code in such a way that different basis set options can be added to PynPoint in the future if desired. The basis set information is stored in the `pdf_basis` attribute of the basis instance as a NumPy array.

The `residuals` class is designed to take data from `images` and `basis` instances and to perform the steps that involve: (i) modelling the PSF; (ii) subtracting the PSF from each frame; (iii) rotating the ADI images to a common sky orientation; (iv) averaging over the stack; and (v) smoothing the images. The most critical input for these calculations is the number of basis functions that should be used to model the PSF. Since the image manipulation steps can be time consuming, we have built in a feature that will store the PSF model information when a given number of basis coefficients is used. This means that any subsequent residual method calls that use the same number of basis functions can by-pass the computationally intensive step and use the previously calculated result directly. However, if the user does a computation that changes the number of basis coefficients, the function will perform a new calculation to model the PSF.

The `basis`, `images`, `residuals` and `workflow` classes offer a `save` feature, which stores all the data of the instances. This allows us to then reconstitute an object using a `restore` feature. For instance, in the following example `images` and `images2` will contain the same information:

```
>>> images.save(filename)
>>> images2 = PynPoint.images.create_restore(filename)
```

The data produced and stored in various instances can be accessed directly and plotted using functionality of the `matplotlib` library. We have also included a plotting module that allows users to easily produce some of the most common figures in a standard way. As an illustration, the panels of Figure 2 can be produced as follows:

```
>>> from PynPoint import PynPlot
>>> PynPlot.plt_im_arr(images, 0)
>>> PynPlot.plt_psf_basis(basis, 1)
>>> PynPlot.plt_res(res, 20)
```

The plotting function `plt_res` accepts a number of keywords such as ‘smooth’, which can be used to smooth the final images through a convolution with a Gaussian of a specified width, and ‘extra\_rot’, which can be used to rotate the images so that (for instance) North points up. The keyword options available to the users are described in our online documentation. The `PynPlot` module can also be used to output a fits file containing the data plotted. This is useful in case users want to examine the images with other packages such as DS9<sup>5</sup>.

`PynPlot` also includes functions for showing animations through the image stack using the `anim_im_arr` function.

<sup>5</sup><http://ds9.si.edu>

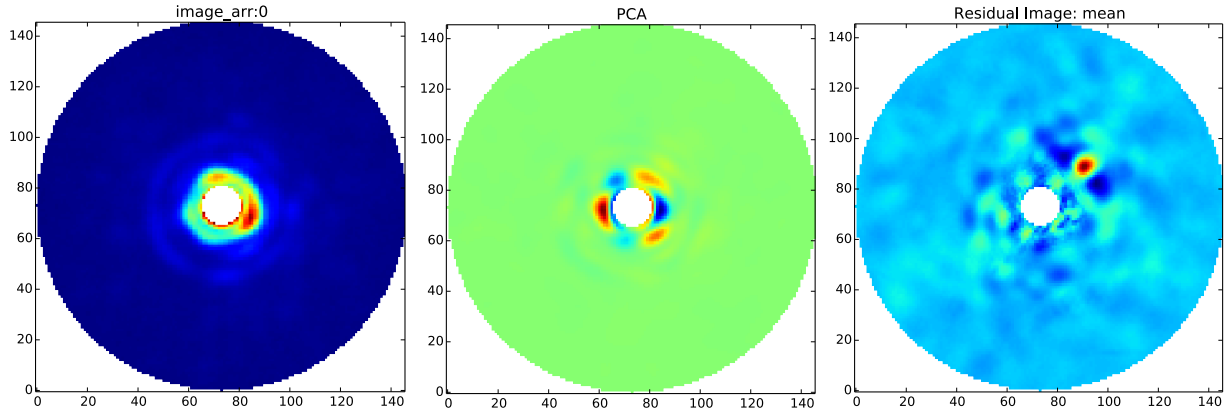


Figure 2: Results plotted using the `pynplot` module. The panel on the left shows one of the 2D images stored in `images`. A mask has been applied (centre and corners), and the images have been normalised and the mean image through the stack has been subtracted. The central panel shows the second principal component stored in `basis`. Finally, right panel shows the final image after correction using the first 20 principal components. The red object to the top-right of centre is the exoplanet  $\beta$ -pic b. In all panels, the colour scale is in arbitrary units.

#### 4.2. Using the workflow engine

Given that users may wish to process several data sets with a number of options, relying exclusively on an interactive mode may not always be ideal. For this reason, we have developed a simple workflow engine for managing PynPoint runs. The `workflow` class takes in a configuration file, where the users can specify all the operations that should be run along with keyword options. Figure 3 shows the design of a PynPoint configuration file. It allows users to define a set of `images` and `basis` configurations, each may have a custom options definition, and a set of `residuals` configurations referring to other definitions. An example of such a config file is shown in Appendix A and further examples can be found in the examples directory of the package.

Dealing with data volumes can be one of the most difficult challenges in analysing a series of target stars. To gain speed, the simplest approach is to load all the data into memory. However, there is a risk that the data volumes can become too large, compared to the available RAM. In this situation, computer systems may begin to do memory swapping to the disk, which causes a catastrophic slowing of calculations.

To allow for variability in the data generated by the workflow, the engine has a flexible design. A key feature of this design is to reduce the coupling between the executed modules. Therefore we introduced the concept of a context object, which adds an abstraction layer to the way the data is being accessed and stored. This level of separation will allow us to easily add more sophisticated features to manage large runs, such as caching, without having to interfere with the main PynPoint functionality. This will yield better backward compatibility, as well as robustness of the implementation.

Similar steps to the ones executed in the interactive example above can be performed using the config file in Appendix A as follows,

```
>>> import PynPoint
>>> ws = PynPoint.run(configfile, force_replace=True)
>>> res = ws.get('residuals_module3')
```

These operations will lead to a residual instance that can then be manipulated and plotted in the usual way. In performing the calculations, `workflow` creates a workspace (`ws`), where all the important information about the run is stored. Setting the `force_replace` option to `True` means that previous versions of the workspace with the same name, if they exist, will be replaced. The default setting (`force_replace = False`) will raise an error if the `workdir` stated in the config file already exists. Knowing the workspace directory (`workdir = dir_ws`), it is possible to restore and recover the intermediate information:

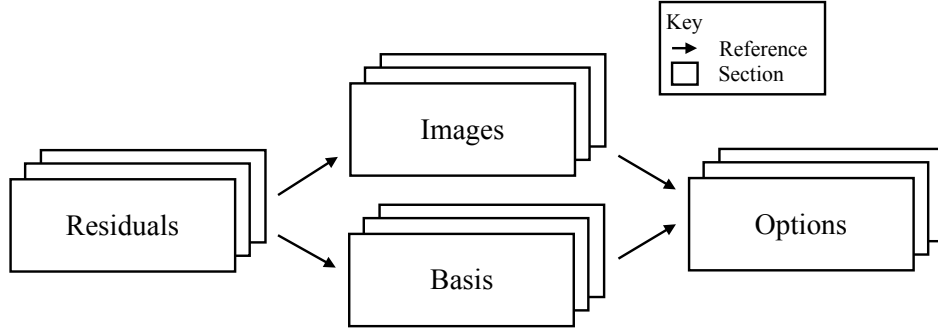


Figure 3: The flexible design of the workflow configuration files allows for customizing the behaviour of the `images`, `basis` and `residuals` instances. We show that a given config file can consist of a large number of sections. Several `residual` instances can be generated, where each one takes inputs from an `images` and a `basis` instance. The sections associated with each `images` and `basis` calculation themselves take their inputs from sections that define the user specified options.

```
>>> import PynPoint
>>> ws = PynPoint.restore(dir_ws)
>>> images = ws.get('images_module1')
>>> basis = ws.get('basis_module2')
>>> res = ws.get('residuals_module3')
```

Here, the instances of `images`, `basis` and `res` have been recovered using the `get` method. The available instances in `ws` can be printed

```
>>> ws.get_available()
```

One further advantage of working with the workflow manager is that communicating the data analysis steps to the rest of the science community becomes simple. Authors need only to include a copy of their PynPoint configuration file and the steps taken will be transparent. If the data is also made available, then the results can be easily reproduced.

#### 4.3. Command line execution of the workflow engine

It is often useful to be able to launch jobs from the command line. This is usually the case when dealing with a queueing system on a computing cluster. To simplify this process, we have also provided a command line interface that can run the workflow engine in a terminal by passing the name of the config file to be used.

```
$ PynPoint <configfile> True
```

Including the `True` at the end has the affect of setting the `force_replace` option to `True`. Once completed, the results can be loaded in the same way as previously introduced.

## 5. Performance

In the following section we report on the performance of the PynPoint package. To do this, we first define our benchmark data set. For continuity and consistency, we have decided to process the same data that we had used to report the science performance of our PCA method [27]. These data were initially used to confirm the massive planet orbiting  $\beta$  Pictoris [14]. The dataset was taken with the VLT NACO instrument on 26 December 2009. The observations were made in the  $L'$  filter in ADI mode, and 80 data cubes, each with 300 individual exposures, were saved. The field rotation over the observations was roughly  $44^\circ$ . The initial data reduction steps, corresponding to the steps described in Section 2, can be found in [22]. As part of the re-centering step, the resolution of the images was doubled. Therefore, the final images used in PynPoint ( $146 \times 146$  pixels) correspond to the central ( $73 \times 73$ ) detector



pixels around the central star. After the removal of bad images, 23,839 images remain corresponding to a total data volume (in .fits files) of 3.2 Gbs.

We ran a bench mark calculation using the configuration and options shown in Appendix A. These were run on a Mac Pro with an Intel Xeon E5 Ivy Bridge 12-core (2.7 GHz) and 64GB (4x16GB) of 1866MHz DDR3 memory. The results are shown in Figure 4. The blue region on the left shows the time spent on calculations of the `images` module and the red region shows the time for the `basis` module. Following this, there is a small (not visible) period while the `residual` instance is being set up. The final phase, in cyan, tidies up the results and outputs the results to disk. We see that the calculations presented here take roughly 1.5 minutes of wall time to complete using `stack_ave = 5` and that the most computationally intensive parts of the calculations are inside the basis phase, where the principal components are being calculated. The principal component computation relies on the SciPy function `linalg.svd` that, as can be seen by the figure, is automatically parallelised through Intel MKL<sup>6</sup> and is able to use multiple cores.

To achieve this run time of 1.5 minutes, we have relied on the fact that we load all of the relevant data into Python and we try to maximise our use of vector and matrix operations. While being fast, the downside of this approach is that the memory demands can be high. In the run we show the memory peak is around 6 Gb. On workstations this should not be a problem, even for large datasets, but smaller machines - such as laptops - will need to run less demanding jobs (for example by increasing the number of frames in a stack). For comparison, a similar run with `stack_ave = 50`, which should offer reasonable results [32], takes a wall time of 3 second and a peak memory load of 600 Mb. On the other hand, it is also possible to perform the calculations with no stacking. For this example, this was done within 40 minutes with a peak memory of roughly 30Gb, which can be accommodated on powerful workstations. In future versions of PynPoint, we will aim to reduce the memory consumption by improving the memory management, which should make large jobs even easier to run.

For the residual calculations, we load the results that are calculated from the workflow run (using the `ws.get` command). Calculating the final residual image for a given number of basis coefficients takes roughly 20 seconds the first time. We have implemented a rudimentary caching scheme so that if the user wishes to calculate other properties from residual files with the same number of basis coefficients, then the calculations are much faster (i.e.  $< 0.01$  seconds for a repeat calculation).

## 6. Quality Assurance

In order to assure a high level of code quality, we have adopted a test driven development methodology, where unit tests are written early in the code development process, ideally before the main code has been written. This helps us write cleaner and well-tested software and ensures that newly developed features do not affect previously written code. The PynPoint unit tests have been developed with the `pytest` testing suite<sup>7</sup>. Much effort has been put into maintaining a high level of test coverage with a focus on ensuring the numerical correctness of the output. Additionally, we tried to test the functionality and interface exposed to the user by taking a best effort approach and ensure that functions test the inputs they are given through assert commands. Furthermore we have also adopted a continuous integration approach. Allowing for automated execution of unit tests and reporting after check in of new functionality ensuring a stable state of PynPoint in our code versioning system at any time. We do this by using the publicly available software Jenkins<sup>8</sup>.

## 7. Summary

We have given a description and brief performance illustration of our Python package PynPoint. This package has been written to analyse imaging data for the direct searches of exoplanets. The current version of the code relies on a principal component analysis method to model the point spread function of the central star. This then allows us to subtract the flux of the star and to search the images for faint companions. We have shown that for typical calculations and data volumes, the PynPoint steps can be calculated on the time scale of roughly 1.5 minutes.

---

<sup>6</sup><https://software.intel.com/en-us/intel-mkl>

<sup>7</sup> <http://pytest.org>

<sup>8</sup><http://jenkins-ci.org>

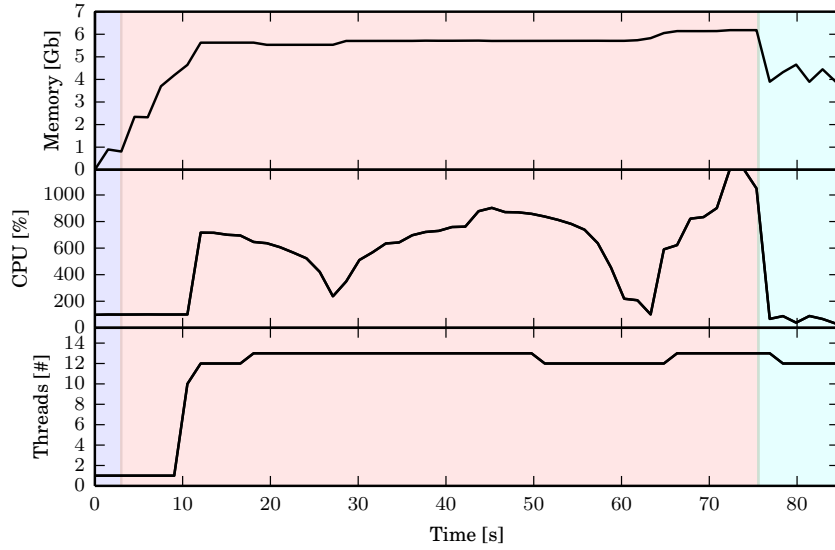


Figure 4: Performance profile for our benchmark calculation. The top-panel shows the memory used in Gb; the middle panel shows the percentage cpu with 100% corresponding one core, and the bottom panel shows the number of threads. The different coloured regions show the various phases of the workflow. From left to right, these calculations are for: `images` (blue), `basis` (red), `residuals` (green - not visible) and the phase where the data is being saved to the disk (cyan). The entire calculation is completed in 1.5 minutes.

We are distributing the code through the central PyPi server<sup>9</sup>, which makes our package easy to access and install. We will also maintain documentation for the package at read-the-docs (see <http://pynpoint.ethz.ch>), which we also use as the project website. As well as the static content delivered through the website, we have also created a wiki<sup>10</sup> for more dynamic content. Users are encouraged to join the PynPoint mailing list ([pynpoint@lists.phys.ethz.ch](mailto:pynpoint@lists.phys.ethz.ch)), which can be done by following the instructions on the PynPoint website. Users keen to contribute to the project should feel free to contact us. Contact details can be found on the PynPoint websites.

## Acknowledgements

We would like to thank Alexander Refregier, Lukas Gamper and Laurenz Gamper for their support through the ETH Cosmology Software Lab. We also thank Michael R. Meyer, Henning Avenhaus, Conor Francois and Maddalena Reggiani for useful discussions in the development and testing of the current release of PynPoint.

<sup>9</sup><https://pypi.python.org/pypi/PynPoint>

<sup>10</sup><https://wiki.phys.ethz.ch/PynPoint>

## Appendix A. Config file example

```
[workspace]
workdir = ../data/workspace_betapic_stk5/
datadir = ../data/

[module1]
mod_type = images
input = Data_beta_pic_L_Band/
intype = dir
options = options1

[module2]
mod_type = basis
input = Data_beta_pic_L_Band/
intype = dir
options = options1

[module3]
mod_type = residuals
intype = instances
images_input = module1
basis_input = module2

[options1]
cent_remove = True
cent_size = 0.05
edge_size = 1.0
resize = False
F_final = 2 # Not used if resize is not True
recent = False
para_sort = True
stackave = 5
```

## References

- [1] J. Schneider, C. Dedieu, P. Le Sidaner, R. Savalle, I. Zolotukhin, Defining and cataloging exoplanets: the exoplanet.eu database, *A&A* 532 (2011) A79. [arXiv:1106.0586](#), [doi:10.1051/0004-6361/201116713](#).
- [2] J. T. Wright, O. Fakhouri, G. W. Marcy, E. Han, Y. Feng, J. A. Johnson, A. W. Howard, D. A. Fischer, J. A. Valenti, J. Anderson, N. Piskunov, The Exoplanet Orbit Database, *PASP* 123 (2011) 412–422. [arXiv:1012.5676](#), [doi:10.1086/659427](#).
- [3] N. M. Batalha, J. F. Rowe, S. T. Bryson, T. Barclay, C. J. Burke, D. A. Caldwell, J. L. Christiansen, F. Mullally, S. E. Thompson, T. M. Brown, A. K. Dupree, D. C. Fabrycky, E. B. Ford, J. J. Fortney, R. L. Gilliland, H. Isaacson, D. W. Latham, G. W. Marcy, S. N. Quinn, D. Ragozzine, A. Shporer, W. J. Borucki, D. R. Ciardi, T. N. Gautier, III, M. R. Haas, J. M. Jenkins, D. G. Koch, J. J. Lissauer, W. Rapin, G. S. Basri, A. P. Boss, L. A. Buchhave, J. A. Carter, D. Charbonneau, J. Christensen-Dalsgaard, B. D. Clarke, W. D. Cochran, B.-O. Demory, J.-M. Desert, E. Devore, L. R. Doyle, G. A. Esquerdo, M. Everett, F. Fressin, J. C. Geary, F. R. Girouard, A. Gould, J. R. Hall, M. J. Holman, A. W. Howard, S. B. Howell, K. A. Ibrahim, K. Kinemuchi, H. Kjeldsen, T. C. Klaus, J. Li, P. W. Lucas, S. Meibom, R. L. Morris, A. Prša, E. Quintana, D. T. Sanderfer, D. Sasselov, S. E. Seader, J. C. Smith, J. H. Steffen, M. Still, M. C. Stumpe, J. C. Tarter, P. Tenenbaum, G. Torres, J. D. Twicken, K. Uddin, J. Van Cleve, L. Walkowicz, W. F. Welsh, Planetary Candidates Observed by Kepler. III. Analysis of the First 16 Months of Data, *ApJS* 204 (2013) 24. [arXiv:1202.5852](#), [doi:10.1088/0067-0049/204/2/24](#).
- [4] W. J. Borucki, D. G. Koch, G. Basri, N. Batalha, T. M. Brown, S. T. Bryson, D. Caldwell, J. Christensen-Dalsgaard, W. D. Cochran, E. DeVore, E. W. Dunham, T. N. Gautier, III, J. C. Geary, R. Gilliland, A. Gould, S. B. Howell, J. M. Jenkins, D. W. Latham, J. J. Lissauer, G. W. Marcy, J. Rowe, D. Sasselov, A. Boss, D. Charbonneau, D. Ciardi, L. Doyle, A. K. Dupree, E. B. Ford, J. Fortney, M. J. Holman, S. Seager, J. H. Steffen, J. Tarter, W. F. Welsh, C. Allen, L. A. Buchhave, J. L. Christiansen, B. D. Clarke, S. Das, J.-M. Désert, M. Endl, D. Fabrycky, F. Fressin, M. Haas, E. Horch, A. Howard, H. Isaacson, H. Kjeldsen, J. Kolodziejczak, C. Kulesa, J. Li, P. W. Lucas, P. Machalek, D. McCarthy, P. MacQueen, S. Meibom, T. Miquel, A. Prsa, S. N. Quinn, E. V. Quintana, D. Ragozzine, W. Sherry, A. Shporer, P. Tenenbaum, G. Torres, J. D. Twicken, J. Van Cleve, L. Walkowicz, F. C. Witteborn, M. Still, Characteristics of Planetary Candidates Observed by Kepler. II. Analysis of the First Four Months of Data, *ApJ* 736 (2011) 19. [arXiv:1102.0541](#), [doi:10.1088/0004-637X/736/1/19](#).
- [5] M. Mayor, M. Marmier, C. Lovis, S. Udry, D. Ségransan, F. Pepe, W. Benz, J. Bertaux, F. Bouchy, X. Dumusque, G. Lo Curto, C. Mordasini, D. Queloz, N. C. Santos, The HARPS search for southern extra-solar planets XXXIV. Occurrence, mass distribution and orbital properties of super-Earths and Neptune-mass planets, *ArXiv e-prints* [arXiv:1109.2497](#).
- [6] A. W. Howard, G. W. Marcy, J. A. Johnson, D. A. Fischer, J. T. Wright, H. Isaacson, J. A. Valenti, J. Anderson, D. N. C. Lin, S. Ida, The Occurrence and Mass Distribution of Close-in Super-Earths, Neptunes, and Jupiters, *Science* 330 (2010) 653–. [arXiv:1011.0143](#), [doi:10.1126/science.1194854](#).
- [7] A. Cumming, R. P. Butler, G. W. Marcy, S. S. Vogt, J. T. Wright, D. A. Fischer, The Keck Planet Search: Detectability and the Minimum Mass and Orbital Period Distribution of Extrasolar Planets, *PASP* 120 (2008) 531–554. [arXiv:0803.3357](#), [doi:10.1086/588487](#).
- [8] G. W. Marcy, H. Isaacson, A. W. Howard, J. F. Rowe, J. M. Jenkins, S. T. Bryson, D. W. Latham, S. B. Howell, T. N. Gautier, III, N. M. Batalha, L. Rogers, D. Ciardi, D. A. Fischer, R. L. Gilliland, H. Kjeldsen, J. Christensen-Dalsgaard, D. Huber, W. J. Chaplin, S. Basu, L. A. Buchhave, S. N. Quinn, W. J. Borucki, D. G. Koch, R. Hunter, D. A. Caldwell, J. Van Cleve, R. Kolbl, L. M. Weiss, E. Petigura, S. Seager, T. Morton, J. A. Johnson, S. Ballard, C. Burke, W. D. Cochran, M. Endl, P. MacQueen, M. E. Everett, J. J. Lissauer, E. B. Ford, G. Torres, F. Fressin, T. M. Brown, J. H. Steffen, D. Charbonneau, G. S. Basri, D. D. Sasselov, J. Winn, R. Sanchis-Ojeda, J. Christiansen, E. Adams, C. Henze, A. Dupree, D. C. Fabrycky, J. J. Fortney, J. Tarter, M. J. Holman, P. Tenenbaum, A. Shporer, P. W. Lucas, W. F. Welsh, J. A. Orosz, T. R. Bedding, T. L. Campante, G. R. Davies, Y. Elsworth, R. Handberg, S. Hekker, C. Karoff, S. D. Kawaler, M. N. Lund, M. Lundkvist, T. S. Metcalfe, A. Miglio, V. Silva Aguirre, D. Stello, T. R. White, A. Boss, E. Devore, A. Gould, A. Prsa, E. Agol, T. Barclay, J. Coughlin, E. Brugamyer, F. Mullally, E. V. Quintana, M. Still, S. E. Thompson, D. Morrison, J. D. Twicken, J.-M. Désert, J. Carter, J. R. Crepp, G. Hébrard, A. Santerne, C. Moutou, C. Sobeck, D. Hudgins, M. R. Haas, P. Robertson, J. Lillo-Box, D. Barrado, Masses, Radii, and Orbits of Small Kepler Planets: The Transition from Gaseous to Rocky Planets, *ApJS* 210 (2014) 20. [arXiv:1401.4195](#), [doi:10.1088/0067-0049/210/2/20](#).
- [9] F. Fressin, G. Torres, D. Charbonneau, S. T. Bryson, J. Christiansen, C. D. Dressing, J. M. Jenkins, L. M. Walkowicz, N. M. Batalha, The False Positive Rate of Kepler and the Occurrence of Planets, *ApJ* 766 (2013) 81. [arXiv:1301.0842](#), [doi:10.1088/0004-637X/766/2/81](#).
- [10] C. D. Dressing, D. Charbonneau, The Occurrence Rate of Small Planets around Small Stars, *ApJ* 767 (2013) 95. [arXiv:1302.1647](#), [doi:10.1088/0004-637X/767/1/95](#).
- [11] J.-L. Beuzit, M. Feldt, K. Dohlen, D. Mouillet, P. Puget, F. Wildi, L. Abe, J. Antichi, A. Baruffolo, P. Baudoz, A. Boccaletti, M. Carbillet, J. Charton, R. Claudi, M. Downing, C. Fabron, P. Feautrier, E. Fedrigo, T. Fusco, J.-L. Gach, R. Gratton, T. Henning, N. Hubin, F. Joos, M. Kasper, M. Langlois, R. Lenzen, C. Moutou, A. Pavlov, C. Petit, J. Pragt, P. Rabou, F. Rigal, R. Roelfsema, G. Rousset, M. Saisse, H.-M. Schmid, E. Stadler, C. Thalmann, M. Turatto, S. Udry, F. Vakili, R. Waters, SPHERE: a planet finder instrument for the VLT, in: *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, 2008. [doi:10.1117/12.790120](#).
- [12] B. A. Macintosh, J. R. Graham, D. W. Palmer, R. Doyon, J. Dunn, D. T. Gavel, J. Larkin, B. Oppenheimer, L. Saddlemyer, A. Sivaramakrishnan, J. K. Wallace, B. Bauman, D. A. Erickson, C. Marois, L. A. Poeyner, R. Soummer, The Gemini Planet Imager: from science to design to construction, in: *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, Vol. 7015 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, 2008. [doi:10.1117/12.788083](#).
- [13] S. Seager, D. Deming, Exoplanet Atmospheres, *ARAA* 48 (2010) 631–672. [arXiv:1005.4037](#), [doi:10.1146/annurev-astro-081309-130837](#).
- [14] A.-M. Lagrange, M. Bonnefoy, G. Chauvin, D. Apai, D. Ehrenreich, A. Boccaletti, D. Gratadour, D. Rouan, D. Mouillet, S. Lacour, M. Kasper, A Giant Planet Imaged in the Disk of the Young Star  $\beta$  Pictoris, *Science* 329 (2010) 57–. [arXiv:1006.3314](#), [doi:10.1126/science.1187187](#).
- [15] C. Marois, B. Macintosh, T. Barman, B. Zuckerman, I. Song, J. Patience, D. Lafrenière, R. Doyon, Direct Imaging of Multiple Planets Orbiting the Star HR 8799, *Science* 322 (2008) 1348–. [arXiv:0811.2606](#), [doi:10.1126/science.1166585](#).

- [16] C. Marois, B. Zuckerman, Q. M. Konopacky, B. Macintosh, T. Barman, Images of a fourth planet orbiting HR 8799., *Nature* 468 (2010) 1080–1083. [arXiv:1011.4918](#).
- [17] J. Rameau, G. Chauvin, A.-M. Lagrange, T. Meshkat, A. Boccaletti, S. P. Quanz, T. Currie, D. Mawet, J. H. Girard, M. Bonnefoy, M. Kenworthy, Confirmation of the Planet around HD 95086 by Direct Imaging, *ApJL* 779 (2013) L26. [arXiv:1310.7483](#), [doi:10.1088/2041-8205/779/2/L26](#).
- [18] M. Kuzuhara, M. Tamura, T. Kudo, M. Janson, R. Kandori, T. D. Brandt, C. Thalmann, D. Spiegel, B. Biller, J. Carson, Y. Hori, R. Suzuki, A. Burrows, T. Henning, E. L. Turner, M. W. McElwain, A. Moro-Martín, T. Suenaga, Y. H. Takahashi, J. Kwon, P. Lucas, L. Abe, W. Brandner, S. Egner, M. Feldt, H. Fujiwara, M. Goto, C. A. Grady, O. Guyon, J. Hashimoto, Y. Hayano, M. Hayashi, S. S. Hayashi, K. W. Hodapp, M. Ishii, M. Iye, G. R. Knapp, T. Matsuo, S. Mayama, S. Miyama, J.-I. Morino, J. Nishikawa, T. Nishimura, T. Kotani, N. Kusakabe, T.-S. Pyo, E. Serabyn, H. Suto, M. Takami, N. Takato, H. Terada, D. Tomono, M. Watanabe, J. P. Wisniewski, T. Yamada, H. Takami, T. Usuda, Direct Imaging of a Cold Jovian Exoplanet in Orbit around the Sun-like Star GJ 504, *ApJ* 774 (2013) 11. [arXiv:1307.2886](#), [doi:10.1088/0004-637X/774/1/11](#).
- [19] S. P. Quanz, A. Amara, M. R. Meyer, M. A. Kenworthy, M. Kasper, J. H. Girard, A Young Protoplanet Candidate Embedded in the Circumstellar Disk of HD 100546, *ApJL* 766 (2013) L1. [arXiv:1302.7122](#), [doi:10.1088/2041-8205/766/1/L1](#).
- [20] M. Reggiani, S. P. Quanz, e. al., in preparation.
- [21] M. A. Kenworthy, S. P. Quanz, M. R. Meyer, M. E. Kasper, R. Lenzen, J. L. Codona, J. H. Girard, P. M. Hinz, An apodizing phase plate coronagraph for VLT/NACO, in: Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, Vol. 7735 of Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, 2010. [arXiv:1007.3448](#), [doi:10.1117/12.856811](#).
- [22] S. P. Quanz, M. R. Meyer, M. A. Kenworthy, J. H. V. Girard, M. Kasper, A. Lagrange, D. Apai, A. Boccaletti, M. Bonnefoy, G. Chauvin, P. M. Hinz, R. Lenzen, First Results from Very Large Telescope NACO Apodizing Phase Plate: 4  $\mu$ m Images of The Exoplanet  $\beta$  Pictoris b, *ApJL* 722 (2010) L49–L53. [arXiv:1009.0538](#), [doi:10.1088/2041-8205/722/1/L49](#).
- [23] D. Mawet, O. Absil, C. Delacroix, J. H. Girard, J. Milli, J. O’Neal, P. Baudoz, A. Boccaletti, P. Bourget, V. Christiaens, P. Forsberg, F. Gonté, S. Habraken, C. Hanot, M. Karlsson, M. Kasper, J.-L. Lizon, K. Muzic, R. Olivier, E. Peña, N. Slusarenko, L. E. Tacconi-Garman, J. Surdej, L-band AGPM vector vortex coronagraph’s first light on VLT/NACO. Discovery of a late-type companion at two beamwidths from an F0V star, *A&A* 552 (2013) L13. [arXiv:1304.0930](#), [doi:10.1051/0004-6361/201321315](#).
- [24] C. Marois, D. Lafrenière, R. Doyon, B. Macintosh, D. Nadeau, Angular Differential Imaging: A Powerful High-Contrast Imaging Technique, *ApJ* 641 (2006) 556–564. [arXiv:arXiv:astro-ph/0512335](#), [doi:10.1086/500401](#).
- [25] D. Lafrenière, C. Marois, R. Doyon, D. Nadeau, É. Artigau, A New Algorithm for Point-Spread Function Subtraction in High-Contrast Imaging: A Demonstration with Angular Differential Imaging, *ApJ* 660 (2007) 770–780. [arXiv:arXiv:astro-ph/0702697](#), [doi:10.1086/513180](#).
- [26] D. Lafrenière, R. Doyon, C. Marois, D. Nadeau, B. R. Oppenheimer, P. F. Roche, F. Rigaut, J. R. Graham, R. Jayawardhana, D. Johnstone, P. G. Kalas, B. Macintosh, R. Racine, The Gemini Deep Planet Survey, *ApJ* 670 (2007) 1367–1390. [arXiv:0705.4290](#), [doi:10.1086/522826](#).
- [27] A. Amara, S. P. Quanz, PYNPOINT: an image processing package for finding exoplanets, *MNRAS* 427 (2012) 948–955. [arXiv:1207.6637](#), [doi:10.1111/j.1365-2966.2012.21918.x](#).
- [28] R. Soummer, L. Pueyo, J. Larkin, Detection and Characterization of Exoplanets and Disks Using Projections on Karhunen-Loève Eigenimages, *ApJL* 755 (2012) L28. [arXiv:1207.4197](#), [doi:10.1088/2041-8205/755/2/L28](#).
- [29] R. Galicher, C. Marois, B. Macintosh, T. Barman, Q. Konopacky, M-band Imaging of the HR 8799 Planetary System Using an Innovative LOCI-based Background Subtraction Technique, *ApJL* 739 (2011) L41. [arXiv:1107.0967](#), [doi:10.1088/2041-8205/739/2/L41](#).
- [30] S. P. Quanz, M. A. Kenworthy, M. R. Meyer, J. H. V. Girard, M. Kasper, Searching for Gas Giant Planets on Solar System Scales: VLT NACO/APP Observations of the Debris Disk Host Stars HD172555 and HD115892, *ApJL* 736 (2011) L32. [arXiv:1106.4528](#), [doi:10.1088/2041-8205/736/2/L32](#).
- [31] S. P. Quanz, J. R. Crepp, M. Janson, H. Avenhaus, M. R. Meyer, L. A. Hillenbrand, Searching for Young Jupiter Analogs around AP Col: L-band High-contrast Imaging of the Closest Pre-main-sequence Star, *ApJ* 754 (2012) 127. [arXiv:1205.6890](#), [doi:10.1088/0004-637X/754/2/127](#).
- [32] T. Meshkat, M. A. Kenworthy, S. P. Quanz, A. Amara, Optimized Principal Component Analysis on Coronagraphic Images of the Fomalhaut System, *ApJ* 780 (2014) 17. [arXiv:1310.8577](#), [doi:10.1088/0004-637X/780/1/17](#).